Cyclical Staff Scheduling Using Constraint Logic Programming

Peter Chan 1 and Georges Weil 2

1 COSYTEC S.A., 4 Rue Jean Rostand, Parc Club Orsay-Université, F-91893 Orsay-Cedex, France
2 Laboratory TIMC-IMAG, Université J. Fourier, Faculté de Médecine de Grenoble, F-38700 Grenoble, France
{pchan|gweil}@equitime.com
http://www.equitime.com

Abstract. In organizations where duty is around the clock, seven days a week and every week of the year, timetabling is a very difficult task, juggling between the workload and the constraints to be respected. Our work concerns cyclical timetabling. This is not just duplicating a fixed sequence of assignments, but has to consider fixed annual leave, and various regulations on assignments on successive days. In some cases, the cycle sequence has to be relaxed and cycle length shortened or extended. In other cases, a small change in leave dates is allowed, except in summer. This paper describes the context and the use of work cycles in the real world, proposes an abstract model to take into account the various constraints, and finally shows how to implement an effective solution using constraint logic programming (in particular, CHIP V5) to produce timetables of up to 150 people over a yearly horizon.

1 Introduction

Employee timetabling addresses the management problem of scheduling human resources every day to meet estimated requirements such that goals and constraints of the management, labor union and personnel are satisfied. In establishments which operate around the clock and all around the year, this is a very highly combinatorial task. Hospitals have an additional dimension of qualification or grade (full diploma nurses, health workers, etc.). In most other cases, the staff is polyvalent and there is no distinction on qualification: police, prison guards, firemen, postal centers or public utility servicemen in the public sector. Private establishments include facility management centers, telephone call centers, tollbooths, bus drivers, etc. Employee timetabling does not simultaneously consider the planning of resources like classrooms or specific equipment.

* Both authors are currently at EQUITIME S.A., 4 Ave. de l’Obiou, F-38700 La Tronche, France.
1.1 Motivation

Timetables can either be cyclic or not. Non-cyclic timetables are created to meet daily requirements that may change every day of the month, yet cope with staff unavailability due to sickness, while satisfying management goals and individual preferences.

Cyclic timetables are based on predefined work cycles also called rotating schedules. They are designed to provide the ideal balance between daily work and rest, and are applied systematically to assure both equality among workers and the best service quality. When the same assignments are made in turn, everybody shares both popular and unpopular work equally.

The work reported in this paper is aimed at generating timetables for groups of some 150 people over one year, based on the choice of a work cycle and usual annual leave patterns. This simulation allows workers to choose their work cycle, after studying in detail resulting timetables, instead of some “theoretical” cycle properties which may be meaningless to workers. The simulator had to produce timetables that satisfy management requirements.

This work follows earlier work of the authors [2], [10] on non-cyclical timetables. Although non-cyclical timetables are outside the scope of this paper, a comparison between the two types of timetables will be given at the end of the paper.

1.2 Contribution

The contribution of this paper is to (a) describe the constraints applicable in timetables based on work cycles, (b) use a model unifying daily and weekly cycles and (c) demonstrate the application of constraint programming (CP) techniques to solve practical problem instances involving 150 people over a period of one year.

Much attention has been concentrated on the generation of work cycles [3], [7]; they are simply duplicated identically over the time horizon. In our users’ establishments, this practice is unacceptable since absences are planned in advance. In addition to meeting daily manpower requirements, work schedules have to be adapted around fixed annual leave or professional training, extended or shortened to allow days off while ensuring rest after night duty. Similar to many practical applications, the real challenge is to introduce relaxation in an acceptable way. This is why CP was used for this application.

The paper continues with details of the cyclic timetable problem in Section 2. Our approach is given in Section 3, followed by a comparison and conclusion in Section 4.

As an anecdote, in one particular institution where a 12 week cycle was proposed, a four month timetable for 15 persons (with some leave periods) was produced manually. It was found that the cycle was disrupted for at least two persons every week, not to mention the severe headache it gave to the planning staff. Today, our application can generate annual timetables without breaking the cycle and in less than 20 s on an Intel Pentium 233 MHz PC.

In this paper, a resource can either be an individual worker or a team of workers who share the same timetable (we will not discuss here the situations where the individual timetable differs from the team timetable in some 20% of the days). The code